Printer With Document-Triggered Processing

Inventors:
Jonathan J. Hull
Kurt Piersol
Peter E. Hart

CROSS-REFERENCES TO RELATED APPLICATIONS

[0001] The present application claims priority under 35 U.S.C. § 119(e) to:

- 1) U.S. Provisional patent application entitled "Printer Including One or More Specialized Hardware Devices" filed on September 25, 2003, having serial number 60/506,303, and
- 2) U.S. Provisional patent application entitled "Printer Including Interface and Specialized Information Processing Capabilities" filed on September 25, 2003, having serial number 60/506,302.

Both provisional applications are herein incorporated by reference in their entirety.

[0002] The following patent applications are related to the present application and are incorporated by reference herein in their entirety:

1) U.S. Patent Application No. ___________, filed March 30, 2004, of Hart et al., entitled "Printer Having Embedded Functionality for Printing Time-Based Media," attorney docket number 20412-08340.

2) U.S. Patent Application No. __________, filed March 30, 2004, of Hart et al., entitled " Printer User Interface," attorney docket number 20412-08455.

3) U.S. Patent Application No. _________, filed March 30, 2004, of Hart et al., entitled "Printer With Hardware and Software Interface for Media Devices," attorney docket number 20412-08383.

BACKGROUND OF THE INVENTION

1. Field of the Invention

[0003] The present invention relates to document printers and, more specifically, to systems and methods that can detect characteristics or content of a printed document and trigger an action in response.

2. Description of the Background Art

[0004] Conventional printers receive documents in a variety of formats and print the contents of the documents in accordance with a proper format. For example, a printer enabled to print PostScript documents will correctly interpret Postscript commands within a document so that the document has the appearance expected by its author when it is printed. (Postscript is a trademark of Adobe Systems Incorporated).

[0005] Today, as more and more electrical databases and computer networks are interconnected, people often have multiple data systems and destinations in which to store information. For example, a person may receive an email containing information that he wants to retain. The person may want to print some or all of the information. The person may further want to add the information to a database or to send the information to other people or destinations or to add the information to a web page. Currently, the person will need to execute several different software programs and will need to type multiple commands into the programs. He may also need to re-enter the information into one or more programs. This is not efficient and is prone to human error, since human beings occasionally forget to perform one of more of the tasks usually associated with a received document and are also prone to typographical errors.

[0006] Some conventional printers incorporate a management function in which the printer monitors its own internal functions and "phones home" for assistance if, for example, its toner is low or it is out of paper. This action is based on the printer doing "self-monitoring," not on any monitoring of the documents to be printed.

[0007] What is needed is a printer and method that can take various actions (in addition to printing) depending on the content and characteristics of a document that is sent to it.

SUMMARY OF THE INVENTION

[0008] The present invention overcomes the deficiencies and limitations of the prior art by providing a system and method for a printer that can detect characteristics of a printed document and trigger an action in response, in addition to or instead of the action of printing the document. For example, such a printer could extract the fields and values in an electronic form and store that information in a database. This system can also create a paper document that reflects some aspect of the action executed by the printer.

[0009] The action is preferably based on content of the document received by the printer. This content can include printable and non-printable data. In certain embodiments, the document is analyzed by a data processing system before it is sent to be printed and additional data is added to or coupled with the document. In one embodiment, an author or other user provides additional information, such as the name of a database. In another embodiment, software on the data processing system analyzes the document and provides additional information about its content, such as the content of certain data fields. In yet another embodiment, the document is analyzed after the printer receives it. In all of these embodiments, the printer performs an action (in addition to or instead of printing) in accordance with the content of the document.

[0010] In certain embodiments, the printer and/or an associated printer-side component interact with the user or with the document source before the printer performs the action in accordance with the content of the document.

BRIEF DESCRIPTION OF THE DRAWINGS

[0011] The invention is illustrated by way of example, and not by way of limitation in the figures of the accompanying drawings in which like reference numerals refer to similar elements.

[0012] Fig. 1(a) is a block diagram showing a system usable in connection with the present invention.

[0013] Fig. 1(b) is a block diagram showing a system usable in connection with the present invention.

[0014] Fig. 1(c) is a block diagram showing a system usable in connection with the present invention.

[0015] Fig. 2 shows an embodiment of the present invention for printing a PDF form and making an entry based on content of the form into a database.

[0016] Fig. 3(a) shows a first embodiment of the present invention for printing a paper document and additionally performing an action.

[0017] Fig. 3(b) shows a second embodiment of the present invention for printing a paper document and additionally performing an action.

[0018] Fig. 3(c) shows a third embodiment of the present invention for printing a paper document and additionally performing an action.

[0019] Fig. 3(d) shows a fourth embodiment of the present invention for printing a paper document and additionally performing an action.

[0020] Fig. 4 shows an example of interactive communication with a printer in accordance with the present invention.

[0021] Fig. 5(a) is a flowchart corresponding to an embodiment of Fig. 3(a).

[0022] Fig. 5(b) is a flowchart corresponding to an embodiment of Fig. 3(b).

[0023] Fig. 5(c) is a flowchart corresponding to an embodiment of Fig. 3(c).

[0024] Fig. 5(d) is a flowchart corresponding to an embodiment of Fig. 3(d).

[0025] Fig. 5(e) is a flowchart corresponding to an embodiment of the present invention.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

[0026] A method and apparatus for performing at least one action (besides printing) based on the content of a printed document is described. In the following description, for purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the invention. It will be apparent, however, to one skilled in the art that the invention can be practiced without these specific details. In other instances, structures and devices are shown in block diagram form in order to avoid obscuring the invention.

[0027] Reference in the specification to "one embodiment," "certain embodiments" or "an embodiment" means that a particular feature, structure, or characteristic described in connection with the embodiment is included in at least one embodiment of the invention. The appearances of the phrase "in one embodiment" in various places in the specification are not necessarily all referring to the same embodiment.

[0028] Some portions of the detailed descriptions that follow are presented in terms of algorithms and symbolic representations of operations on data bits within a computer memory. These algorithmic descriptions and representations are the means used by those skilled in the data processing arts to most effectively convey the substance of their work to others skilled in the art. An algorithm is here, and generally, conceived to be a self-consistent sequence of steps leading to a desired result. The steps are those requiring physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated. It has proven convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers, or the like.

[0029] It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities. Unless specifically stated otherwise as apparent from the following discussion, it is appreciated that throughout the description, discussions utilizing terms such as "processing" or "computing" or "calculating" or "determining" or "displaying" or the like, refer to the action and processes of a computer system, or similar electronic computing device, that manipulates and transforms data represented as physical (electronic) quantities within the computer system's registers and memories into other data similarly represented as physical quantities within the computer system memories or registers or other such information storage, transmission or display devices.

[0030] The present invention also relates to apparatus for performing the operations herein. This apparatus may be specially constructed for the required purposes, or it may comprise a general-purpose computer selectively activated or reconfigured by a computer program stored in the computer. Such a computer program may be stored in a computer readable storage medium, such as, but is not limited to, any type of disk including floppy disks, optical disks, CD-ROMs, and magnetic-optical disks, read-only memories (ROMs), random access memories (RAMs), EPROMs, EEPROMs, magnetic or optical cards, or any type of media suitable for storing electronic instructions, and each coupled to a computer system bus.

[0031] The algorithms and displays presented herein are not inherently related to any particular computer or other apparatus. Various general-purpose systems may be used with programs in accordance with the teachings herein, or it may prove convenient to construct more specialized apparatus to perform the required method steps. The required structure for a variety of these systems will appear from the description below. In addition, the present invention is not described with reference to any particular programming language. It will be appreciated that a variety of programming languages and Page Description Languages (PDLs) may be used to implement the teachings of the invention as described herein.

[0032] Moreover, the present invention is claimed below operating on or working in conjunction with an information system. Such an information system as claimed may be the entire messaging system as detailed below in the preferred embodiment or only portions of such a system. Thus, the present invention is capable of operating with any information system from those with minimal functionality to those providing all the functionality disclosed herein. Fig. 1(a) is a block diagram showing a system usable in connection with the present invention. A document having document content 105 is sent from a document source 102 to a printer 106. As used herein, the term "content" refers to anything in the print stream sent to the printer, including both printing and non-printing data. In the Figure, the document is sent over a network 104, such as the Internet, an intranet, a wireless connection, a wide area network, or the like. Printer 106 receives the document and performs an action based on the content of the document. One example of such an action is to send data to a component 108. In certain embodiments, printer 106 also prints the document 110. The action performed may be any of a number of actions, such as entering data in a database, sending a notification or confirmation, adding data to a web page, etc. Examples of actions performed in accordance with content of the document are discussed in further detail below.

[0033] Fig. 1(b) is a block diagram showing a system usable in connection with the present invention. In this example, document source 102 is connected to printer 106 without a network connection. Similarly, printer 106 is connected to component 108 without a network connection.

[0034] Fig. 1(c) is a block diagram showing a system usable in connection with the present invention. In this example, printer 106 is connected to component 108 using a network connection through network 104. Network 104 can be any network, such as the Internet, an intranet, a wireless connection, a wide area network, or the like.

[0035] It will be understood that the system configurations shown in Figs. 1(a)-1(c) are examples only and are included to show some configurations usable with the present invention. It will be understood that other configurations are possible. For example, the connections between the document source and the printer and between the printer and component 108 can both be network connections.

[0036] Fig. 2 shows an embodiment of the present invention in which a printer 204 prints 206 a PDF form 202 and also performs an action, such as making an entry based on content of the form 202 into a database 208. For example, if form 202 is a form filled out by a user, the form is printed 206 and data in fields of the form are added to the user's record in database 208. In certain embodiments, fields of the document are extracted using a well-known API for Acrobat plug-ins that allows access to the fields of PDF files. In certain other embodiments, fields are extracted from HTML or XML documents using well-known techniques.

[0037] In this example, printer 204 communicates with database 208 over a network, although printer 204 could also have a direct connection to database 208. Database 208 can be external or internal to printer 204. In this example, printer 204 includes network interface hardware and software 212 required to communicate over a network. Printer 204 further includes Internet communication software 210 that creates and executes commands to communicate with the database. These commands are, for example, SQL commands, http commands, or similar commands, depending on the nature of database 208.

[0038] Figs. 3(a)-3(c) show further implementations of a printer that receives a document and performs an action in accordance with content of the document. It will be understood that the components of these figures can be connected with or without a network connection and that any connection method enabling the communication described below falls within the sprit and scope of the present invention.

<u>Author-provided Data, Application Plug-in, Simple Document Processing by Printer</u>

[0039] Fig. 3(a) shows a first embodiment of the present invention for printing a paper document 306 and additionally performing an action 308. Fig. 5(a) shows a flow chart of a method used by this system. In Fig. 3(a), a data processing system 302 communicates with a printer 304. Data processing system 302 includes an initial copy of the document to be printed 322. This document 322 can have any appropriate format, including but not limited to ACSII text, Postscript, Microsoft Word, Adobe's pdf format, etc. This document optionally includes author-provided data, as discussed below (see 502). Data processing system 302 also includes a document rendering application 324, such as Acrobat Reader (available from Adobe Systems Incorporated). In this example, document rendering application 324 includes an application plug-in 325. Application plug-in 325, which may be supplied with a print driver, inspects every document printed by document rendering application 324 and preferably performs the following (see 504):

a. determines whether the document needs special processing and whether the document includes author-provided data, e.g., an identifier for a database that receives the content of a form (Note that "author-provided data" can be provided by any user with the capability to augment the document to be printed, not just the author of the document. For this reason, it is also referred to as "user-provided data".); and

b. optionally applies processing, e.g., field extraction from forms or URL extraction from web pages, to produce application-derived data; and

c. communicates author-provided data and application-derived data to the printer in a way that does not affect the appearance of the printed document (see 506).

[0040] Data processing system 302 then sends information to the printer using a format such as Postscript (see 508). This information is termed "Page Description Language" (PDL) because it describes the appearance and characteristics of a document to be printed. In this example, the PDL includes both author provided data (such as an identifier for a database that receives the content of a form) and data derived by application plug-in 325 (such as data extracted by the plug-in from fields in the document). It will be understood that the functionality of plug-in 325 could also be implemented as part of software 324, instead of as a plug-in.

[0041] The communication of data to the printer in a way that does not affect the appearance of the printed document described in step (c) above can be done by including the data as a non-printing comment field in the PDL file. This type of non-printing comment is currently allowed in Adobe Postscript or Hewlett-Packard's PCL. Often a long sequence of unlikely characters signals the presence of such data (see "Postscript Language Document Structuring Conventions Specification, Version 3.0," Sept. 25, 1992, http://partners.adobe.com/asn/developer/pdfs/tn/5001.DSC_Spec.pdf). Using Postscript as an example, in which comments are indicated by two percent signs at the beginning of a line, data fields for a form that indicate the first and last names of an applicant could be indicated as:

```
%%RICOH-NEPV1.0-FORMITEM page=1 field=”First Name” type=”string” value=-“Jonathan”
%%RICOH-NEPV1.0-FORMITEM page=1 field=”Last Name” type=”string” value=-“Hull”
```

[0042] Fig. 3(a) further shows a printer 304 having document processing software 326. In this example, document processing software 326:

a. determines whether the document contains author-provided data, application-derived data, and whether any further processing should be applied to the document;

b. optionally processes the document to produce printer-derived data, e.g., a filter might be applied to postscript data that detects web URL's. In this document, processing by the printer is classified as simple or complex. In general, simple processing involves the printer executing a well-defined external command or recognizing regular expressions (such as URLs). Complex processing by the printer involves performing a content-based operation on the PDL.

In a preferred embodiment, the printer software 326 executes simple processing. It inspects the comments in the PDL file and when it finds:

%%RICOH-NEPV1.0-CONTENT-FILTER NO

it determines that a content-based analysis of the PDL is not needed. Instead, a well-defined external command is executed.

c. decides and perform an action in accordance with the content of the received document, e.g., a pdf form might include author-provided data that specifies a web location for a database and application-derived data that includes form field identifiers and values. In this step the printer creates a database update command from this data and executes it. A simple example of a database update command to update the first and last name fields in a database would be a cgi call such as:

http://www.ca.dmv.gov/special_plate.cgi?last=Hull&first=Jonathan

[0043] In summary, in the system shown in Fig. 3(a), when the user executes a print command, the application processes the source document and creates a page description language (PDL) version of the document that includes the author-provided data and application-derived data computed by an application plug-in. Document processing software on the printer detects the presence of that information and decides whether it should compute additional printer-derived data from the PDL document. It also decides what action should be executed, as a function of the author-provided, application-derived, and printer-derived characteristics of the document.

[0044] In another embodiment, plug-in 325 generates non-printed commands in the print stream in a manner similar to that discussed below in connection with Fig. 3(a). In such an embodiment, the commands are generated by the plug-in, however, rather than by an author or user as discussed below. In various embodiments, plug-in 325 generates commands causing the printer to do one or more of the following:

-Collect all form data from the print stream and insert it into a database;

-Perform arbitrary calculations and sending mail messages to a known address

-Detect web URLs in the document and instruct a company's web proxy server to pre-cache those pages or store an archival copy

-Download web URLs and print those web pages as an attachment to the document being printed.

[0045] Yet another embodiment contains either a modified web browser or a web browser with a plug-in in place of document rendering application 324. The modified browser inserts similar commands when HTML forms are printed.

[0046] In certain embodiments, the printed document 306 is modified to reflect the results of the action performed by the printer. For example, the document may be modified to contain a watermark that certifies that the data printed on the paper is exactly the same as the data that was uploaded into a database. In certain other embodiments, a certificate to this effect is printed along with the document. In certain other embodiments, an email is sent to the document source, document author, or other indicated party to this effect.

[0047] The following paragraphs discuss other implementations of the present invention.

Author-provided Data, No Application Plug-in, Simple Document Processing by Printer

[0048] Fig. 3(b) shows a second embodiment of the present invention for printing a paper document and additionally performing an action. Fig. 5(b) shows a flow chart of a method used by this system. In this example, the user again provides data (such as the name of a database to be updated), but the system does not require a plug-in for document rendering application 324. In this example, the author includes a command in a document to be printed by adding the command as a non-printing comment field in the document. This type of non-printing comment is currently allowed in, for example, Adobe Postscript and Hewlett-Packard's PCL printer language. In an embodiment of the invention, the user includes a command in the comments that indicates that the printer should email him when the document is printed. Thus, the action performed by the printer is the generation and sending of an email message to the user.

[0049] This implementation is useful in systems where data sent to the printer is commonly stored in a print queue. In such systems, the protocol does not allow for user interaction with the printer. Thus, any special communication must be embedded within the data initially sent to the printer. Ideally, the embedded data does not affect the print spooler or other forwarding device, while remaining easy for the printer to detect and parse. As an example, the user inserts the following PostScript metadata into the print stream. Note that this metadata has the format of a non-printing comment.

```
%%RICOH-NEPV1.0-EMAIL-UPON-PRINTING docname="Special Plate Application" time-queued="10:11:37"
dest="hull@rii.ricoh.com"
```

The above example indicates a document name, e.g., "Special Plate Application" and an email address where notification of printing is to be emailed: hull@rii.ricoh.com.

[0050] Document processing software 326 of printer 304 looks for commands within the print stream and processes them. Thus, in the above example, software 326 would send an email message to hull@rii.ricoh.com when this document is printed. Other examples of actions taken by printer 304 in various embodiments include one or more of the following:

-Collect all form data from the print stream and insert it into a database;

-Perform arbitrary calculations and sending mail messages to a known address

-Detect web URLs in the document and instruct a company's web proxy server to pre-cache those pages or store an archival copy

-Download web URLs and print those web pages as an attachment to the document being printed.

No Author-Provided Data, No Application Plug-in, Complex Document Processing by Printer

[0051] Fig. 3(c) shows a third embodiment of the present invention for printing a paper document and additionally performing an action. Fig. 5(c) shows a flow chart of a method used by this system. In such an implementation, analysis of the document is performed on the printer 304, eliminating the need for modifications or plug-ins on the data processing system 302. In one embodiment, the printer analyzes halftone images sent to the printer. This analysis allows the printer to monitor its use of toner. A filter applied to a PDL file by document processing software 326 detects halftone rendering. A running tabulation of half tone printing is maintained on the printer and is periodically reported to an email address (either a predetermined email address or an email address specified in the document). In general, the author name often may be obtained from the document itself. The IP address of the author's computer is normally communicated to the printer with the document to be printed.

[0052] Another embodiment monitors the user of commercial fonts in printed documents and calculates license fees required by user of the fonts. Data about the license fees are optionally stored in a database or printed or emailed to an appropriate location.

[0053] Another embodiment monitors the document for predefined pornographic words (or images, identified by name) and notifies an authority figure if such is found. This embodiment is extremely helpful in monitoring misuse of company resources.

[0054] Another embodiment monitors the document for confidential terms such as project names or passwords and notifies an authority figure if such are found in a document to be printed. This embodiment is extremely useful in helping to implement internal security procedures.

[0055] The above embodiments provide examples of applying a calculation or analysis to the document received by the printer 304 and performing an action (besides printing) based on the calculation or analysis.

No Author-provided Data, Application Plug-in, Simple Document Processing by Printer

[0056] Fig. 3(d) shows a fourth embodiment of the present invention for printing a paper document and additionally performing an action. Fig. 5(d) shows a flow chart of a method used by this system. This implementation includes no author-provided data, but does include a plug-in of similar modification to the document rendering application or browser 324. In one embodiment, document rendering application 324 detects web URLs in a document to be printed and embeds those URLs in the document or its PDL file. Document processing software 326 receives these URLs as part of a command to communicate with a local proxy server. The proxy server is instructed to pre-cache the pages corresponding to the detected URLs, under the assumption that those pages will be accessed shortly after the user reads the printed document. Alternatively, the document processing software 304 retrieves the indicated pages and archives them in a local database. Alternatively, the analysis to detect the URLs is performed by the printer 304 instead of by data processing system 302. As described above, the pages could be cached, stored in a database, or printed in addition to the document.

[0057] In certain embodiments, the plug-in detects and sends all form fields in a document. In certain other embodiments, the plug-in searches for certain fields, either in response to user input or because the names of the forms are pre-programmed.

Additional Embodiments

[0058] It will be understood by persons of ordinary skill in the art that the processing described above can be performed at various locations within the system. For example, the processing described above in connection with "complex document processing" performed by the printer 304 could also be performed, either totally or partially by data processing system 302. Similarly, processing described as being performed by software on data processing system 302 could be performed by document processing software in printer 304. Similarly, processing could be performed both by the data processing system/document source and by the printer (see, for example, flowchart of Fig. 5(e)).

[0059] One advantage of using plug-ins in data processing system 302 is that they are easy to add to existing software. In contrast, one advantage of performing most or all processing on the printer side is that printers are easier to maintain and service than user computers. In addition, if a large enterprise is going to purchase special processing software, it is less expensive to purchase software that runs on a small number of printers as opposed to software that runs on each user machine. This is even more important if special purpose hardware (such as a powerful signal processor) is needed to perform the function, because of the expense and difficulty of such installations. Moreover, the work environment of printers is usually better monitored and it is physically easier to maintain security over printers that contain sensitive analysis programs (such as programs that monitor compliance with licensing or seat requirements or with security requirements) than if this analysis software was located in users' computers.

[0060] The following paragraphs contain additional examples of "complex processing" by a printer. The printer recognizes a PDL file directly and analyzes the file to detect certain known fields. For example, if the printer will be used to print many Department of Motor Vehicle (DMV) forms (i.e., forms having a known format and content), the printer can analyze the printed documents to detect predetermined values such as name, address, etc and construct a database update command. In certain embodiments, the printer asks for user confirmation before updating the database (see following section on interactive printer commands).

[0061] In another example, the printer renders the PDL file as an image using, for example the PDL interpreter on the printer, and applies a forms recognition system such as the Ascent Capture package available from Kofax Image Products of Irvine, California. Again, in certain embodiments, the printer asks for user confirmation before performing the action.

Interactive Communication with a Printer

[0062] Fig. 4 shows an example of interactive communication with a printer in accordance with the present invention.

[0063] In general, conventional printer drivers in modern operating systems are not designed to facilitate interactive information gathering. Because the print job can be redirected to another printer, or the printing protocol does not allow such interactive sessions, the operating system does not encourage interaction with the user. Once initial printer settings are captured, further interactions are generally not allowed in conventional printers. One approach to this problem is to embed metadata into the print stream itself, as noted above. However, it is possible that the printer could need to ask the user for more information, in response to computations made from the data supplied by the user. In addition, the printer itself might delegate some tasks to other application servers, which might in turn need more information from the user. So-called "Web services" or "grid computing" systems are examples of the sort of application server that the printer might trigger.

[0064] In order to allow this interaction, without modifying printer driver architecture of the underlying operating system, an extra mechanism, such as the one shown in Fig. 4, is constructed. A “UI Listener,” program 454 listens to a network socket, accepts requests for information 408, interacts with a user to obtain such data, and then sends the data back to the requester.

[0065] Once a print request 402 is sent by user 450 and notification requested from the UI listener 404, the print job is sent by application 452. Here, the print job contains embedded information including the network address of the UI listener, authentication information, and the latest time that the client will be listening for requests.

[0066] If the printer requires additional information of confirmation, it sends a request 408, which is detected by the UI listener, which displays a dialog box to obtain input from the user 410. An example of such a request might be a request for a password or user confirmation code that the user must enter to access a database 458. Alternatively, the printer asks for additional information based on computation, such as noting the absence of a required field and asking for a value to be entered before printing. The user’s input is included in a reply 412 sent to the printer. If the reply does not satisfy the printer it may ask for additional information (not shown). If the reply does satisfy the printer, it takes a next step. This step might be to perform an action such as sending an email (not shown). The next step might also be sending a request for information 414 to an application server (such as a database) 458. In this example, application server 458 also sends a request for information 416, which is detected by the UI listener. The user is prompted 418 and his response forwarded to the application server 420. In this example, a reply is then sent from the application server 458 to the printer 456. It will be understood that a particular embodiment may include either or none or requests 408 and 416 without departing from the spirit of the present invention.

[0067] A program such as that shown in Fig. 4 may have a fixed set of possible interactions, or may accept a flexible command syntax that allows the requester to display many different requests. An example of such a command syntax would be the standard web browser's ability to display HTML forms. These forms are generated by a remote server, and displayed by the browser, which then returns results to the server. In this embodiment, however, the UI listener is different from a browser in that a user does not generate the initial request to see a form. Instead, the remote machine generates this request. In the described embodiment, the UI listener is a server, not a client.

[0068] Because network transactions of this type are prone to many complex error conditions, a system of timeouts would be necessary to assure robust operation. Normally, each message sent across a network either expects a reply or is a one-way message. Messages which expect replies generally have a timeout, a limited period of time during which it is acceptable for the reply to arrive. In this embodiment, embedded metadata would include metadata about a UI listener that will accept requests for further information. Such metadata preferably includes at least a network address, port number, and a timeout period. It might also include authentication information, designed to prevent malicious attempts to elicit information from the user. Because the user cannot tell whether the request is coming from a printer, a delegated server, or a malicious agent, prudence suggests strong authentication by the UI listener. If the printer or a delegated application server wishes more information, it can use the above noted information to request that the UI listener ask a user for the needed information.

[0069] While the present invention has been described with reference to certain preferred embodiments, those skilled in the art will recognize that various modifications may be provided. For example, certain embodiments employ multiple application servers, acting in cooperation to perform a function or request. Any of the above functions or requests can be sent across a network, or using local cables such as IEEE1394, Universal Serial Bus, or wireless networks such as IEEE 802.11 or IEEE 802.15 networks, in any combination. Variations upon and modifications to the preferred embodiments are provided for by the present invention, which is limited only by the following claims.